Amendments to the Claims

Please amend Claims 1, 3, 8-11, 14-16, 22-25, 28-31, 33, 35-37, 40-41, 43, 48-51, 54-55, and 57, and cancel Claim 56, all as shown below.

1. (Currently Amended) A method for providing a request to a portlet wherein the portlet ~~can render~~ renders itself in a graphical user interface (GUI), comprising:

mapping the request to a control tree wherein the control tree is a logical representation of the GUI and wherein the control tree includes a set of controls representing graphical and functional elements of the GUI which are related hierarchically to one another;

advancing the control tree through at least one life cycle stage based on the request, wherein the control tree includes a portlet control that represents the portlet; and

providing the request to a portlet container, wherein the providing ~~can be~~ is performed by the portlet control.

2. (Original) The method of claim 1, further comprising:

generating the control tree from a factory based on the request.

3. (Currently Amended) The method of claim 1, further comprising:

generating a response wherein the response ~~can be~~ is used to render at least a portion of the GUI.

4. (Original) The method of claim 2 wherein the step of generating a control tree from the factory comprises:

creating a metadata representation of a control tree; and

generating a class to construct the control tree based on the metadata representation.

5. (Original) The method of claim 1 wherein:

the request is a hypertext transfer protocol request (HTTP); and

the request originates from a web browser.

6. (Original) The method of claim 3, further comprising:

providing the response to a web browser.

7. (Original) The method of claim 1 wherein:

the control tree is driven through the at least one lifecycle stage by an interchangeable lifecycle component.

8. (Currently Amended) The method of claim 1 wherein:

each one of the set of controls ~~can have~~ has an interchangeable persistence mechanism.

9. (Currently Amended) The method of claim 1 wherein:

each one of the set of controls ~~can render~~ renders itself according to a theme.

10. (Currently Amended) The method of claim 1 wherein:

each one of the set of controls ~~can communicate~~ communicates with another one of the set of controls.

11. (Currently Amended) The method of claim 1 wherein:

one of the set of controls ~~can advance~~ advances through the at least one lifecycle stage in parallel with another of the at least one controls.

12. (Original) The method of claim 1 wherein:

the at least one lifecycle stage is one of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose.

13. (Original) The method of claim 3 wherein:

the response is an hypertext transfer protocol (HTTP) response.

14. (Currently Amended) The method of claim 1 wherein:

controls ~~can~~ raise events and respond to events.

15. (Currently Amended) The method of claim 1 wherein:

each one of the set of controls ~~can be~~ is one of: Book, Page, Window, Menu, Layout, Portlet, Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons.

16. (Currently Amended) A method for providing a request to a portlet wherein the portlet ~~can render~~ renders itself in a portal graphical user interface (GUI), comprising:

mapping the request to a control tree wherein the control tree is a logical representation of the GUI and wherein the control tree includes a set of controls representing graphical and functional elements of the GUI which are related hierarchically to one another;

advancing the control tree through at least one life cycle stage based on the request, wherein the control tree includes a portlet control that represents the portlet;

providing the request to a portlet container, wherein the providing ~~can be~~ is performed by the portlet control; and

generating a response wherein the response ~~can be~~ is used to render at least a portion of the GUI.

17. (Original) The method of claim 16, further comprising:

generating the control tree from a factory based on the request.

18. (Original) The method of claim 17 wherein the step of generating a control tree from the factory comprises:

creating a metadata representation of a control tree; and

generating a class to construct the control tree based on the metadata representation.

19. (Original) The method of claim 16 wherein:

the request is a hypertext transfer protocol request (HTTP); and

the request originates from a web browser.

20. (Original) The method of claim 16, further comprising:

providing the response to a web browser.

21. (Original) The method of claim 16 wherein:

the control tree is driven through the at least one lifecycle stage by an interchangeable lifecycle component.

22. (Currently Amended) The method of claim 16 wherein:

each one of the set of controls ~~can have~~ has an interchangeable persistence mechanism.

23. (Currently Amended) The method of claim 16 wherein:

each one of the set of controls ~~can render~~ renders itself according to a theme.

24. (Currently Amended) The method of claim 16 wherein:

each one of the set of controls ~~can communicate~~ communicates with another one of the set of controls.

25. (Currently Amended) The method of claim 16 wherein:

one of the set of controls ~~can advance~~ advances through the at least one lifecycle stage in parallel with another of the at least one controls.

26. (Original) The method of claim 16 wherein:

the at least one lifecycle stage is one of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose.

27. (Original) The method of claim 16 wherein:

the response is an hypertext transfer protocol (HTTP) response.

28. (Currently Amended) The method of claim 16 wherein:

controls ~~can~~ raise events and respond to events.

29. (Currently Amended) The method of claim 16 wherein:

each one of the set of controls ~~can be~~ is one of: Book, Page, Window, Menu, Layout, Portlet, Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons.

30. (Currently Amended) A system for providing a request to a portlet wherein the portlet ~~can render~~ renders itself in a graphical user interface (GUI), comprising:

a first container operable to map the request to a control tree factory;

the control tree factory operable to generate a control tree based on the request wherein the control tree ~~can include~~ includes at least one portlet control, wherein the control tree is a logical representation of the GUI, and wherein the control tree includes a set of controls representing graphical and functional elements of the GUI which are related hierarchically to one another;

a lifecycle driver operable to drive the control tree through at least one life cycle stage;

a portlet container operable to accept the request from the at least one portlet control and provide the request to the portlet.

31. (Currently Amended) The system of claim 30 wherein:

the portlet ~~is capable of generating~~ generates a response and wherein the response ~~can be~~ is used to render at least a portion of the GUI.

32. (Original) The system of claim 30 wherein:

the request is a hypertext transfer protocol request (HTTP); and

the request originates from a web browser.

33. (Currently Amended) The system of claim 31 wherein:

the response ~~can be~~ is provided to a web browser.

34. (Original) The system of claim 30 wherein:

the at least one portlet control has an interchangeable persistence mechanism.

35. (Currently Amended) The system of claim 30 wherein:

the at least one portlet control ~~can render~~ renders itself according to a theme.

36. (Currently Amended) The system of claim 30 wherein:

each one of the at least one portlet controls ~~can communicate~~ communicates with another one of the at least one portlet controls.

37. (Currently Amended) The system of claim 16 wherein:

one of the at least one portlet controls ~~can advance~~ advances through the at least one lifecycle stage in parallel with another of the at least one portlet controls.

38. (Original) The system of claim 30 wherein:

the at least one lifecycle stage is one of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose.

39. (Original) The system of claim 31 wherein:

the response is an hypertext transfer protocol (HTTP) response.

40. (Currently Amended) The system of claim 30 wherein:

the at least one portlet control ~~can raise~~ raises events and respond to events.

41. (Currently Amended) A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

map a request to a control tree wherein the control tree is a logical representation of a graphical user interface (GUI) and wherein the control tree includes a set of controls representing graphical and functional elements of the GUI which are related hierarchically to one another;

advance the control tree through at least one life cycle stage based on the request, wherein the control tree includes a portlet control that represents a portlet; and

provide the request to a portlet container, wherein the providing ~~can be~~ is performed by the portlet control.

42. (Original) The machine readable medium of claim 41, further comprising instructions that when executed cause the system to:

generate the control tree from a factory based on the request.

43. (Currently Amended) The machine readable medium of claim 41, further comprising instructions that when executed cause the system to:

generate a response wherein the response ~~can be~~ is used to render at least a portion of the GUI.

44. (Original) The machine readable medium of claim 42, further comprising instructions that when executed cause the system to:

create a metadata representation of a control tree; and

generate a class to construct the control tree based on the metadata representation.

45. (Original) The machine readable medium of claim 41 wherein:

the request is a hypertext transfer protocol request (HTTP); and

the request originates from a web browser.

46. (Original) The machine readable medium of claim 43, further comprising instructions that when executed cause the system to:

provide the response to a web browser.

47. (Original) The machine readable medium of claim 41 wherein:

the control tree is driven through the at least one lifecycle stage by an interchangeable lifecycle component.

48. (Currently Amended) The machine readable medium of claim 41 wherein:

each one of the set of controls ~~can have~~ has an interchangeable persistence mechanism.

49. (Currently Amended) The machine readable medium of claim 41 wherein:

each one of the set of controls ~~can render~~ renders itself according to a theme.

50. (Currently Amended) The machine readable medium of claim 41 wherein:

each one of the set of controls ~~can communicate~~ communicates with another one of the set of controls.

51. (Currently Amended) The machine readable medium of claim 41 wherein:

one of the set of controls ~~can advance~~ advances through the at least one lifecycle stage in parallel with another of the at least one controls.

52. (Original) The machine readable medium of claim 41 wherein:

the at least one lifecycle stage is one of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose.

53. (Original) The machine readable medium of claim 43 wherein:

the response is an hypertext transfer protocol (HTTP) response.

54. (Currently Amended) The machine readable medium of claim 41 wherein:

controls ~~can~~ raise events and respond to events.

55. (Currently Amended) The machine readable medium of claim 41 wherein:

each one of the set of controls ~~can be~~ is one of: Book, Page, Window, Menu, Layout, Portlet, Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons.

56. (Cancelled).

57. (Currently Amended) A system comprising:

a means for mapping the request to a control tree wherein the control tree is a logical representation of the GUI and wherein the control tree includes a set of controls representing graphical and functional elements of the GUI which are related hierarchically to one another;

a means for advancing the control tree through at least one life cycle stage based on the request, wherein the control tree includes a portlet control that represents the portlet; and

a means for providing the request to a portlet container, wherein the providing ~~can be~~ is performed by the portlet control.